

PDE

fields

#	string	fvt	fht	section
0	st storage type	0	F	a
1	nl name length	1	F	
2	nam name	2	7	
3	typ type	4	7	
4	kb key block	5	C	b
5	bu blocks used	6	E	
6	len length	7	9	
7	cre creation	8	B	
8	ver version -- minimum	9	16	c
9	acc access	A	9	
A	aux auxiliary	B	C	
B	mod last modification	C	14	
C	ndb header block	D	X F	d
D	res reserved	<u>(b)</u> 5	B	
E	enl entry length	B	X F	
F	epb entries per block	C	14	
10	fls file count	D	X D	e
11	wt bit map block	E	10	
12	bpu blocks per volume	F	X 14	
13	pbk parent block	E	F	
14	pen parent entry number	F	16	f
15	pel parent entry length	16	16	

idx # (message number)

idx fvt, x

sta vtab

jsr basale

txa

asl

tax

idx fieldtbl, x

sta addr

tax

idx fieldtbl, x

sta addr

idx ftoff

jsr pstr

asto = E2

1028-1068



fs: a b c d

vs: a e c f g

ss: a e c f h

vtc → - - -

fil fids:

vol fids:

sub fids:

to	1	2	3	4	5	6	7	8	9	A	B	C	D
fil fids:	0	1	2	3	4	5	6	7	8	9	A	B	C
vol fids:	0	1	2	3	D	7	8	9	E	F	10	11	12
sub fids:	0	1	2	3	D	7	8	9	E	F	10	13	14 15

vtc (4)

PDE

dspfs_ display field section

a

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
storage type: seedling 25
1 name length: 5 324
2 name: CHRIS 27 10 15 20 25
3 type: PPP 545



b

4 key block:
5 blocks used.
6 length:
7 creation:

date:

dsr date 01/11/11 11:11:11



edit

```

ldx curloff
lda off11,x
sta curaddr
clc
lda curpg
add off1b,x
sta curaddr+1

```

ecout

```

ldz curi
bne
dbz #0
lda (curaddr),y
and #%.ffff0000
cmp #%.11110000
bne
sub
fil

```

exit

i (Curaddr) → star top/ham len

if not at dir. header entry, then edit fi fids

; st/b1

; sz

; not dir bbr

PDE

* = data additional to hex

fields	VIA	FVS	X
st * storage type	0 0 0		0
nl name length	1 1 1		1
name * name	2 2 2		2
type * type	4 4 4		3
kb key block	5		4
bu blocks used	6		5
len length	7		6
cre * creation	8 8 8		7
ver version -> minimum	9 9 9		8
acc * access	A A A		9
aux auxiliary	B		A
mod * last modification	C		B
hdr header block	D		C

* str : address of string start
(term. with 000)

fieldtbl

bits	res	reserved	D
bb	el	entry length	E
BB	eb	entries per block	F
CC	fc	file count	10
DD	fm	bit map block	11
EE	fv	blocks in volume	12
FF	pb	parent block	13
	peh	parent entry number	14
	pel	parent entry length	15

code:

lda #x
asl
tax
lda fieldtbl, x
sta addr
lax
lda fieldtbl, x
sta addr+1
ldy tab
isr prstr

display
Section
subroutines

filflds	(D)	file entry	0 1 2 3 4 5 6 7 8 9 A B C	ABCD
volflds	(D)	vol dir	0 1 2 3 4 5 6 7 8 9 A B C	AECFG
subflds	(E)	sub dir	0 1 2 3 4 5 6 7 8 9 A B C	AECFH
ffvt		f vlt	0 1 2 3 4 5 6 7 8 9 A B C	
vvvt		v vt	0 1 2 3 4 5 6 7 8 9 A B C	
svvt		s vt	0 1 2 3 4 5 6 7 8 9 A B C	

types

typntbl

typstbl

lda filetype
tax #
cmp typntbl, x
beq

lda filetype

sta asto

lda #0

lda typntbl, x

beq

inx

cmp asto

bne

dex

stx asto

tax

asl

clc

adc asto

tax

lda typstbl, x

lax

jsr cout

lax

lda typstbl, x

jsr cout

lax

not found

months

Jan - 0
Feb - 1
Mar - 2
Apr - 3
May - 4
Jun - 5
Jul - 6
Aug - 7
Sep - 8
Oct - 9
Nov - A
Dec - B

monthtbl

lda #x
asl
tax
lda monthtbl, x
sta addr
lax
lda monthtbl, x
sta addr+1

access

roottbl

delete
rename
write
read
write
delete
write
delete
write
delete
write

storage type

sttbl

- 0 deleted
- 1 seedling
- 2 sapling
- 3 tree
- 4 }
- 5 }
- 6 }
- 7 }
- 8 }
- 9 }
- A subdirectory
- E header (subdirectory)
- F header (volume directory)

1002A
9563
9327
234

PDE

storage type: sapling
name length: \$8
name: CALCULTEVOX0000
.....

type: sys

key block: \$xxxx

blocks used: \$xxxx

length: \$xxxxx

258
7 34 58 23 21 35 27 ...

FF
\$8 \$b
\$5 \$b
\$7 \$5 \$b

Created

bits	bytes
2	name length
2	key block
2	blocks used
3	length
2	version → minimum
2	aux type
2	header block
8	reserved
1	entry length
1	entries per block
2	file count
2	bit map block
2	blocks in volume
2	parent block
1	parent entry number
1	parent entry length

bits	bytes
15	name
15	storage type
1	type

bits	bytes
4	created
4	modified

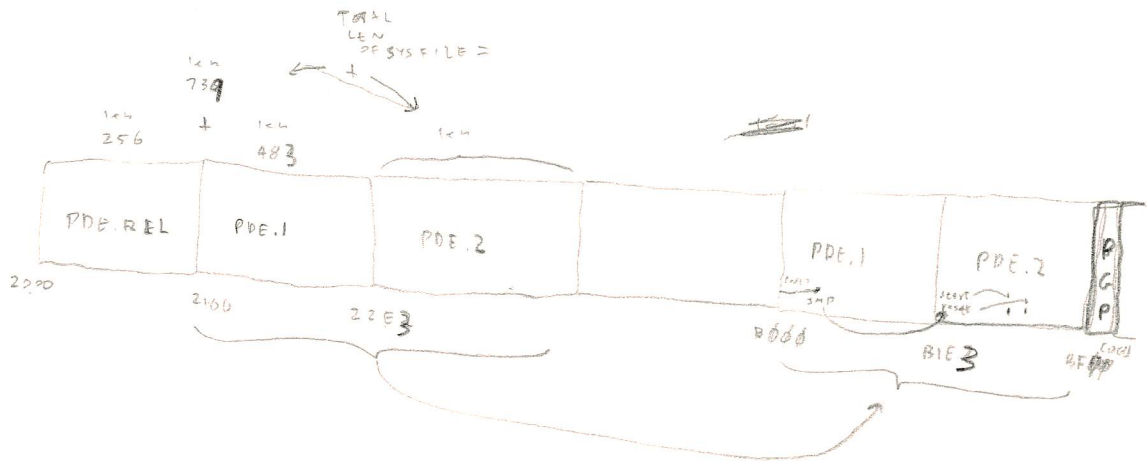
bits	bytes
1	access

sub- / super-directories

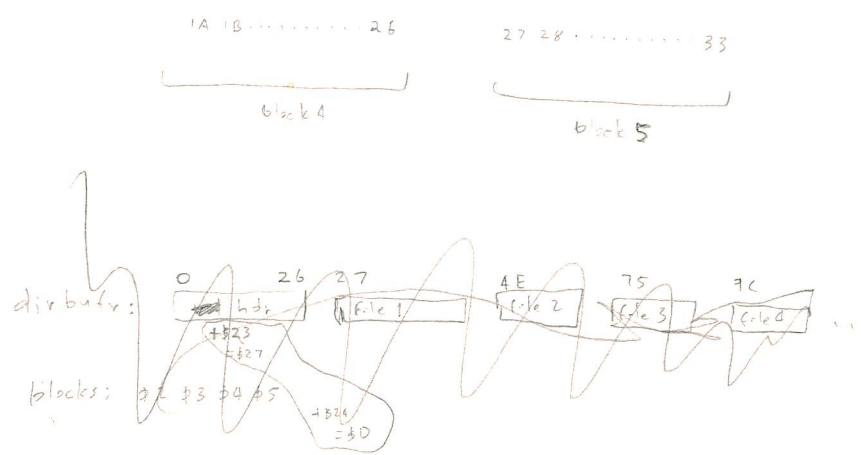
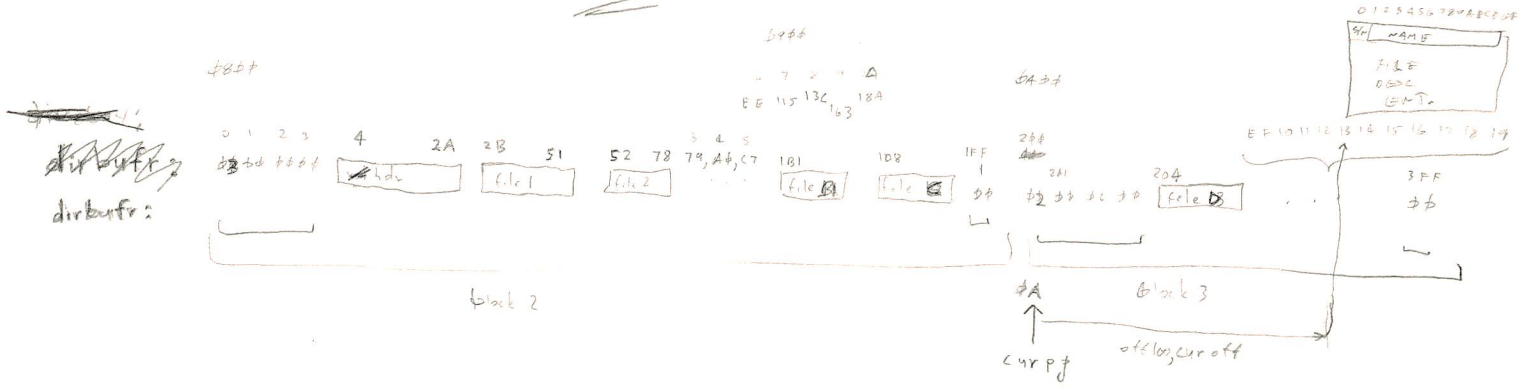
935

addressed
~~in~~ in PDE.1

exit	Bφ3E	vt9b	Eφ
permess	Bφ4A	invmask	E1
cinout	Bφ6D	aslo	E2
cout	Bφ7B	b7s	Fφ
ainvout	BφAφ		
out	BφAE		
inout	BφB8		
mount	BφDA		
bascale	BφF3		
etch	B1φD		
waitkey	B11B		
dsp:ile	B127		
home	B172		
tlhome	B178		
crldown	B17C		
file	B1B4		
start	B1E2		
reset	B1E5		



PDE



```

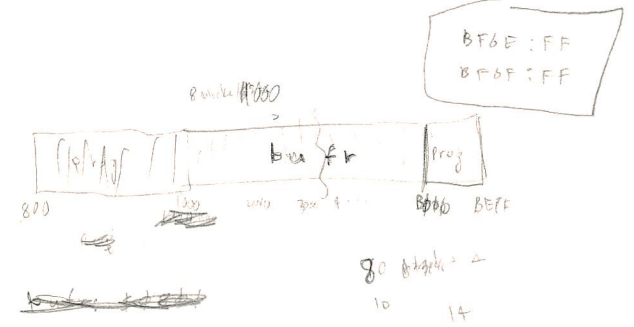
setp lda machid
and #b2
bne ssk1
jmp exit
ssk1 lda #b72
jmp etyentry
    
```

```

exit lda #b15
jsr etyentry
jsr mli
dcb quit
dcb <ex1, >ex1
ex1 dcb b2
dcb b, b, b, b, b
    
```

offh	b4	2B	52	79	A7	C7	EE	15	3C	63	8A	B1	D8
offh	b4	2B	52	79	A7	C7	EE	15	3C	63	8A	B1	D8
+	0	1	2	3	4	5	6	7	8	9	A	B	C

BIMAP ALLOCATION



bufp: b8

84 blocks max
54 blocks max

PDE test

B003	Cur1	0
4	curp0	8
5	marpg	F
6	cur off	0
7	invt	8 = 11
8	topn	7

B1CE: flemenn

B03B: prrws

3	invt	cur off	block
0	1	2	3

B041: dnding

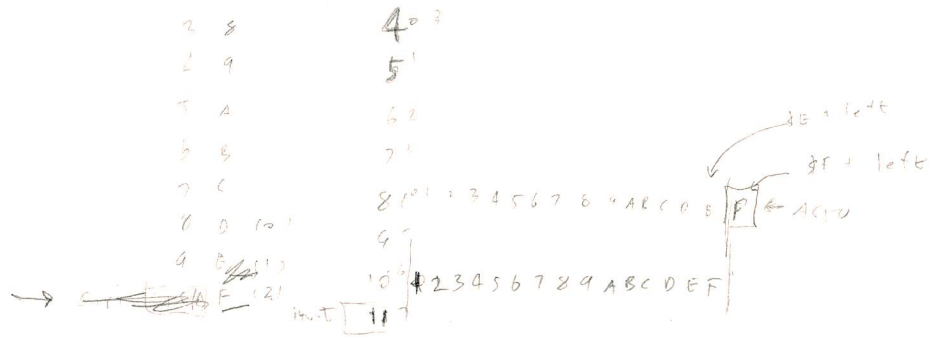
top	bot	11*
0	1	2

B044: off1

B1CE: flemenn

B1FC: dspfwih

B275: dspflin



$$x = 2 - 7 = -5$$

y = 8

$$x = FB$$

$$= 3$$

$$y = 8$$

$$x = 0 \text{ (cur off)}$$

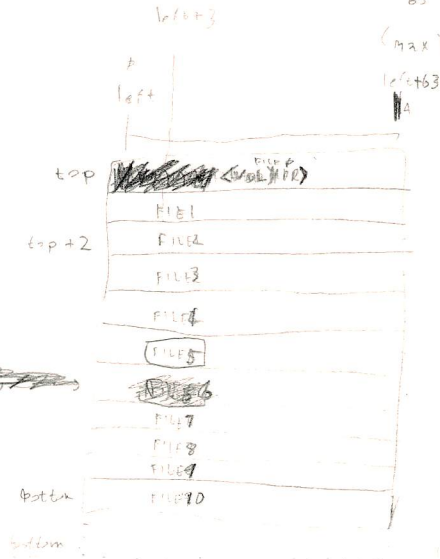
$$y = 8 \text{ (cur off)}$$

PDE

3

63

dividing; top bottom left
4 14 0



sample results

4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

~~inv = top + bottom / 2~~

curi = ~~inv~~ (current inventory)

(value of inventory)
~~inv = top + bottom / 2~~ (= 9)

(minimum value for normal printing)
~~top = 4~~ (= 5)

~~top = 4~~

top = ~~inv - top~~
(number of files above inv) (= 5)

for curi < ~~top~~

vtab = ~~inv~~ - curi - 1
print blank line
print FILE1
↓
print FILE (last) (vtab = bot)

after last file printed,
print blank line (unless
part vtab 23)
if \leq vtab inv, then
cannot move down further

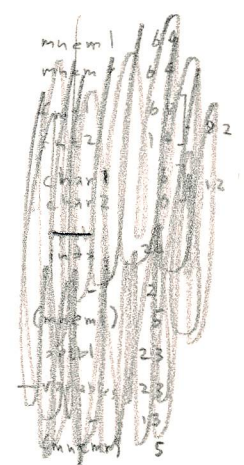
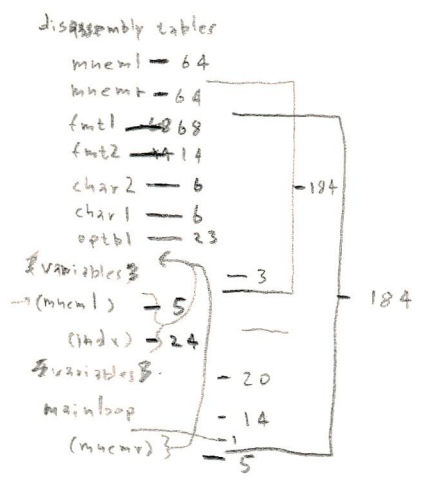
-for other curi; vtab = top
print FILE
↓
curi - top
print FILE (last)

FSMON

title
equates
 opcodes
 page 566 locations
~~page 566 locations~~
 page 566 locations
 rom locations



global table
 org
 jmp table(interface)
 cmd table
 jmp table(codes)



~~init~~
copyaux } init routines
 page
 cbtoads } page printing
 cbtoada
 pline
 memset
 dispmset } memory for
dsrfs - direct header printing
dsprins } instruction printing
prins
 chout } cursor output
 curoff
scrollup } scrolling
scrolldw

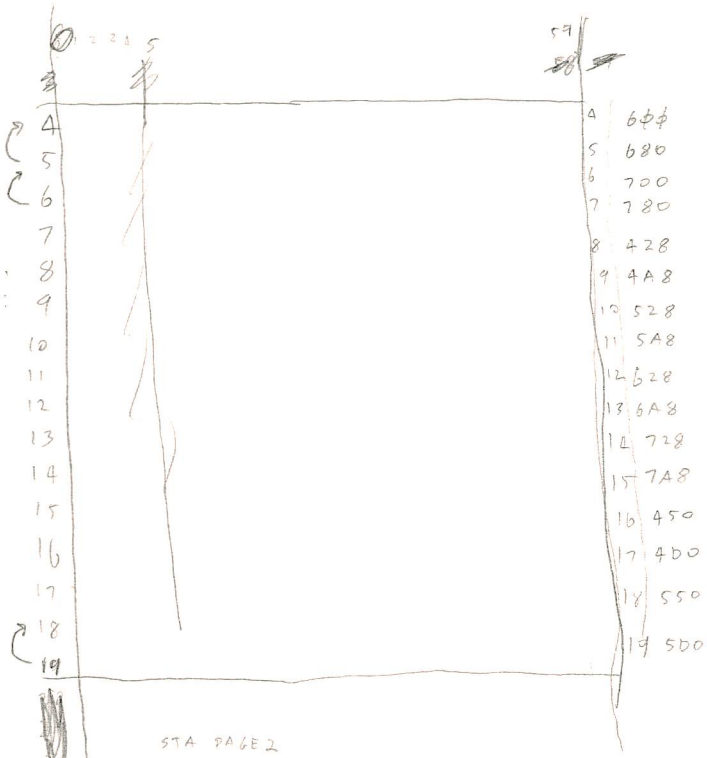
prmess
cinout
cout
 cinout
 cout
 bascale
 fetch
 execute

hex -
up
down
left
right
begin
end
rtog
mtog
exit

row in grid

BLOXAP

SCROLL UP



STA PAGE2

LDY #6

0 ≤ Y ≤ 29, MAIN AND AUX

LDA 3000,Y
STA 3000,Y
LDA 3000,Y
STA 3000,Y

LDA 37500,Y
STA 37500,Y
INY
CPY #30

BCC

LDA PAGE2?

BPL

STA PAGE1

BES

RTS

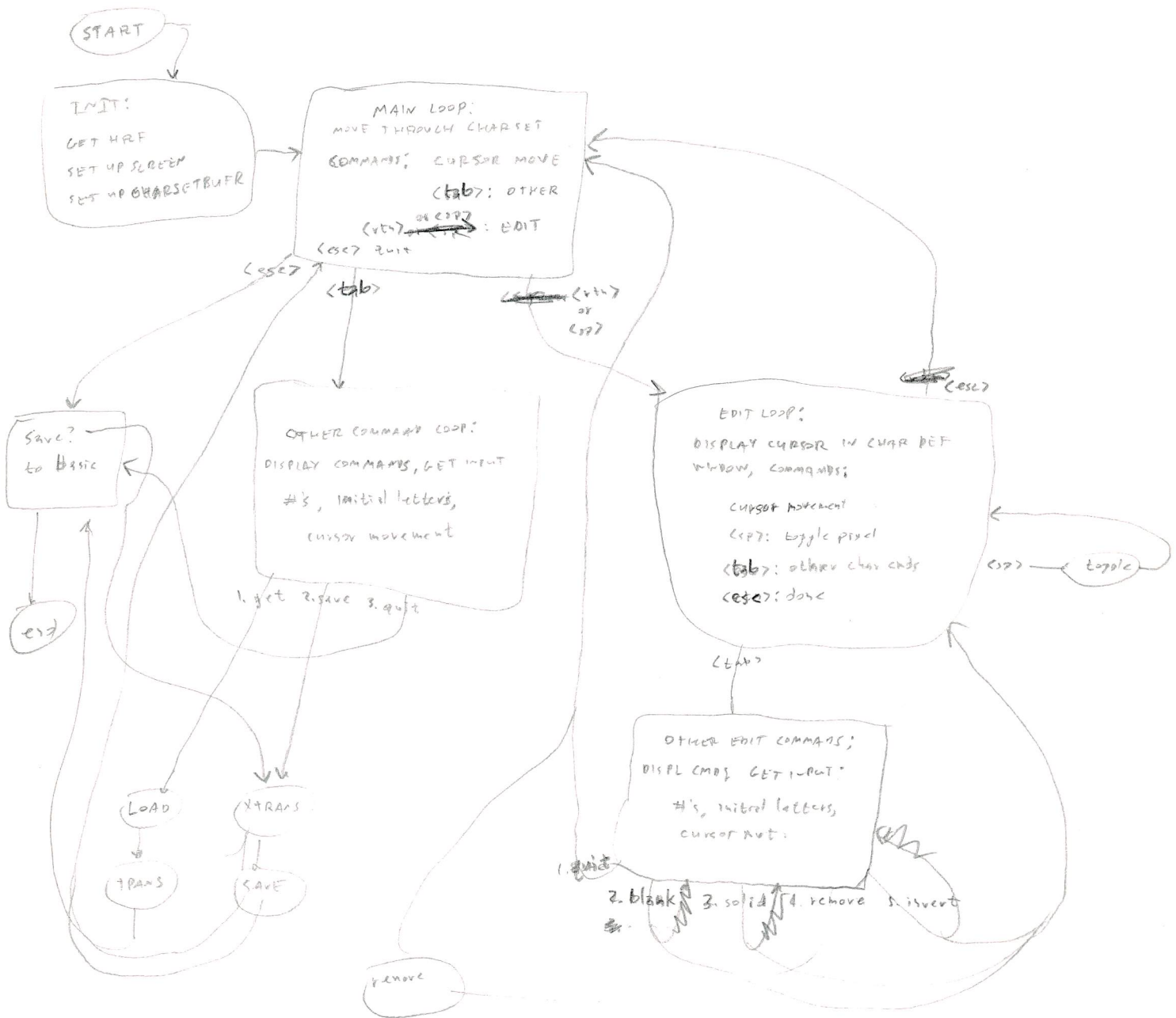
make sure no page boundaries crossed

(less than 30, do next Y)

(no, PAGE1, THEN PAGE, BRANCH)

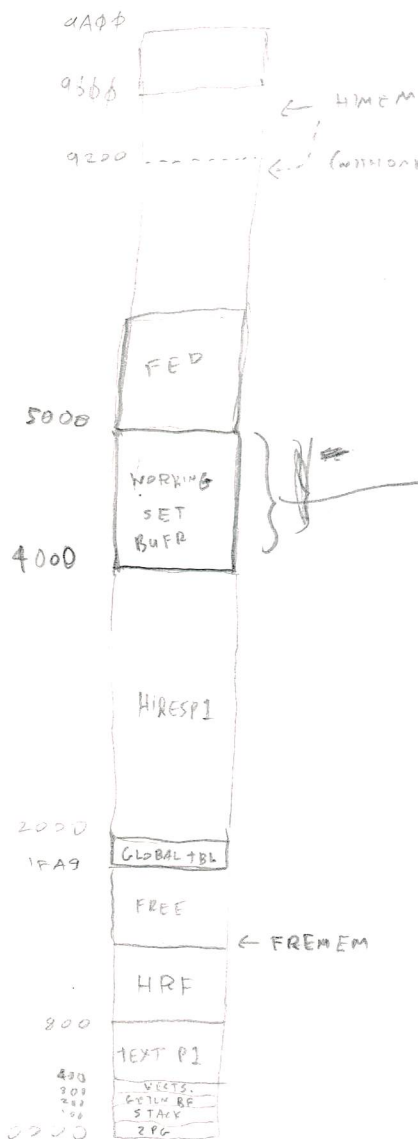
(yes, PAGE2, THEN DO PAGE1)

FED

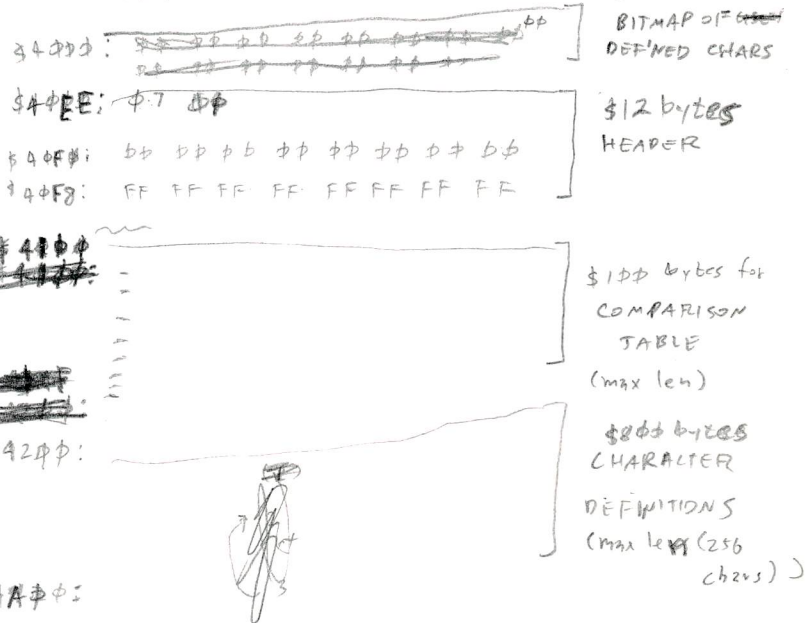


FED

MEMORY MAP



WORKING SET BUFFER:



\$4000 - \$401F: BITMAP
 \$401E - \$40FF: HEADER
 \$4100 - \$41FF: COMPARISON TABLE
 \$4200 - \$49FF: DEFINITION TABLE

FOR I/O → FOR PROG WORKING

$$1.023 \times 1,000,000 \text{ cycles} \times \frac{1 \text{ sec}}{60 \times 60} = \frac{17,050}{17,050} = 17,050$$

17,050 = 429A
 341 x 50
 11 x 31 x 5
 11 x 31 x 5 x 5 x 2
 110 x 155
 152 = 17 x 8

FED

Character set

datatype $\phi 7$ (ignored in 4 ϕ /8 ϕ cols) \leftarrow (CHRADDR)

section
bitmap E4 76543210
 11100000

section sec 0 1 2 3 4 5 6 7
base $\phi\phi$ $\phi\phi$ $\phi\phi$ $\phi\phi$ $\phi\phi$ $\phi\phi$ $\phi\phi$ $\phi\phi$
offsets off 2 3 4 5 6 7 8 9

section sec 0 1 2 3 4 5 6 7
lengths FF FF FF FF FF FF FF FF
(-1)

Comparison

table

section 2

4 ϕ 41 42 43

\leftarrow (CMPADDR)

section 5

section 6

section 7

definitions

FF FF FF FF FF FF FF FF
22 22 22 22 22 22 22 22

FED

(DEFADDR) → 00 01 02 03 04 05 06 07
 ↑
 CURLIN

(defaddr) → 07
 76543210
 E4 (11100100) section bitmap
 0 1 2 3 4 5 6 7
 00 00 00 00 00 20 40 60 section base offsets
 FF FF 1F FF FF 1F 1F 1F section length minus one
 A B C D E F 10 11

12
 11

12

} section 3

(cmpaddr) →

HRF

253
254
075
733
714
1034
1037
1040

LOAD FILE \$800 → \$15EE

COPY CODE TO MBSR02 → \$D000 → \$D0FF

20 cols.
(same version use AX BS02 also)

160 cols
is illegal

INIT:

WINDOW SIZE
DISPLAY SW's

COPY GLOBAL TABLE TO JUST UNDER \$2000

FIX ENV, turns it on: hi bit 0678 off.

CLEAR SCREEN

SET ADDRESS OF PTRS (to start prog at \$4000)

FOR OUTDEV (SETUP ~~SECTOR~~ I/O VECTS (PRODOS or DOS 3.3))

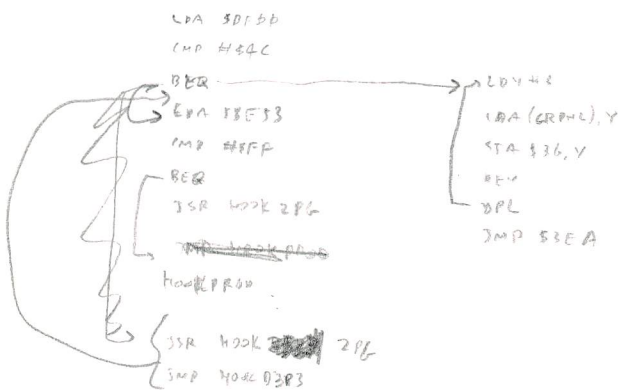
(FOR PRINTER:

→ BE30 - BE33 ← STORE USER VECTORS
IN
→ KOUT2 → CEYIN1

3698

~~SECTOR~~
to INSTALL OWN P/O DEV
SET IFE2 - IFE5 (OUT, IN)
~~JSR~~ 31FFA

MM 0000 → PRGMEM 152E - 1616
AX 0000 → ALTP02 1617 - 1671



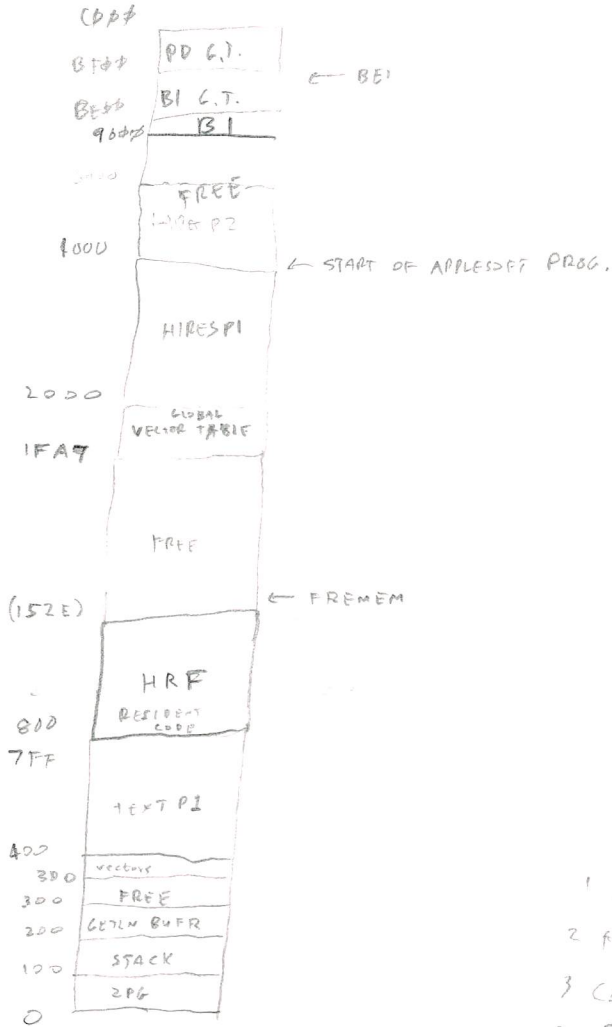
PRO	PRO	3.3
DOS	APP	—
→ 76	→ 3BE30	→ 336
→ BE3D	RTS	
RTS		JMP 33EA

HOOK 2PG
HOOK PROD
HOOK DTPT

(~ (P1=76)) ~ (~ (P2=64))

HRF

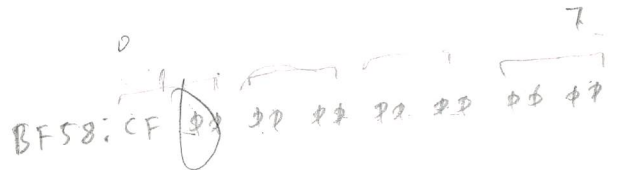
Memory map



W BITMAP:

USED PAGES:

0, 1, 4-3F, 96-BF



1. HRF
2. FIX. DESCEND
3. CONT
4. PED
5. SYM

GLOBAL VECTOR TABLE

\$1FA9:

10 12 14 16 18 1A 1C 1E 10 112

FontADDR 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

FA% = 8105

IFBD > WHEREMEM
IFBE

WAX = 8125

IFBF CHARARG

IFC > FREMEM
IFC

IFC2 ID

IFC3 > ESCCURSR
IFCA

IFCB > ESCCURV
IFCC

IFCD SSCOSIB

IFCE INVERTO

IFCF UNDERCR

IFD0 PAGE

IFD1 > GTLNCHK
IFD4

IFD5 > BELVEC
IFD6

IFD7 > KEYPRVEC
IFD8

IFD9 > TBLIOVEC
IFDE

IFDD > DELAYS
IFDE

IFDF JMP CYCLE

IFE2 JMD SCRNCHR

IFES JMD EXTCLRV

IFE8 JMD DEFADR

IFEB JMD ENVIRON

IFEE JMD BOOTUP

IFF1 JMD BLSTCHR

IFF4 JMD OUTDEV

IFF7 JMD EXTESCE

IFFA JMD EXTESCF

IFFD JMD EXTESCE@ HMY. = 8189

00000000 00000011

A1: 0000

A2: 7371

A3: BE DD

A4: 4000

109C
152D

L\$ 492

L1170

LRA #10
STA A\$L 3C
STA A\$L 42
LRA #14
STA A\$L 3E

LDA #10
STA A\$H 3D
STA A\$H 43
LDA #14
STA A\$H 3F

SEC
JMP \$C311

Small
original
bold
text
uncial
irish

data
blocks of
normal
strings

last byte

* starts with
1 byte: # of addrs;
1st of addrs

* "get ~~addr~~^{addr} pet addr ~~entry~~ ~~from table~~
~~entry~~
i: X, R Z = # above
Y: entry number
where which entry X to print str
get address of Yth 2. " 0: addr "

```
get_coin  
get_coin address  
i: coin  
o: addr → coin start
```

	dn	os	oi	od	rs	ri	rd	ed	nt	dia	com
coin 0:	0	0	0	0	1	1	0	0	0	0	1
coin 1:	1	2	2	1	3	3	1	0	1	1	2

~~As~~ ~~11/11/11~~ ~~11/11/11~~

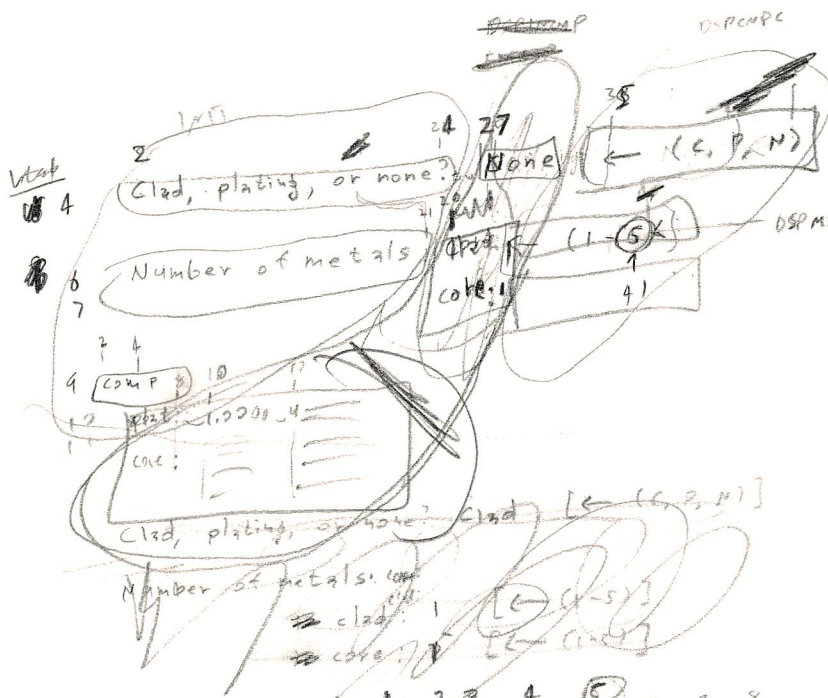
$A = C \quad \beta$

$$\begin{array}{r} 1100 \ 0000 \\ 1000 \ 0000 \\ \hline 100 \ 0000 \\ \hline 1000 \ 0000 \\ \hline 0 \quad 1 \end{array}$$

~~addr. 16~~

addr 2 L0	addr 7 H1
0100 0000	0000 0010
4 6	6 2

$$\text{weight} | + 1 + 3 \times Y$$



55
b - composition:
1.0000 unknown

3
4
5
6
7
8

← # of metals + 2
composition + 3

→ clad: 1.0000 unknown
→ core: 1.0000 unknown

→ Enter the figures: 1.0000

10
11
12
13
14
15
16

ex: C 13 (core) 0000

inc tab
if tab = #metals + 10,
tab := 10

dec tab
if tab = 9,
tab := #metals + 9

2000 0000 → 6
0000 0000 → 5
0000 0010 → 5

201
202 inbuf + 1
203 #2
204 jmp first

692

Choose metal:

→ unknown
copper
steel
manganese
silver
gold

Enter a new metal:

input x = 70

160
cwp
cs

160 #10
cwp
230 5

inbuf: 00 01 00 00 00

dis 4

try
lda inbuf, y
pla
"core:"
pla
jsr without

623
668
688

1FB2
11100100

prevail
i: 2(h...)
over
addr

Some of's after 54

5/5

Memory consumption

field	max string
0	
1	
2	
3	
4	
5	
6	
7	\$2FE
8	\$1FF
9	
A	

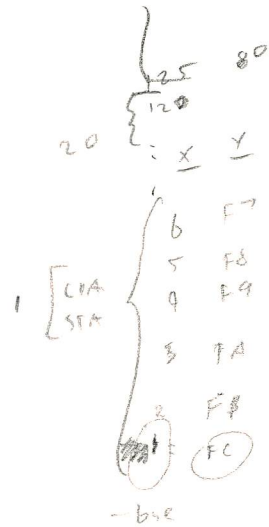
applied pps.

3
2

esc from input
denom

working for adding

den
sub
incs chr ("")
depth
edge ("")
wt
metal



$$Y = FD - X$$

$$Y = 253 - X$$

965

384
392

302-214

AA FF ldr #FFF
 AB bb ldr #0
 AC bb ldr #3
 CD ldr #16
 CE 18 ldr #16
 DD FD ldr #16
 EF ldr #16

$$P_{LONG} = 8A$$

OLD TOP
BIM
LEFT } OK

A

$\begin{array}{r} 10100101 \\ 10000101 \end{array}$

$x = 8A$

 $x = 8A$

(COPYSTART) \rightarrow KEY

2466, Y
5400, Y

2066 DOOU
1
2165 DUFF
2166 blou
1
2265 DUFF

2466Y

~~AD 84 06~~
 29

DOA2L
 137

CD COM
 EF MENU
 G MONITOR
 HI ZERO MENUS
 TL BLOXAP

C-D
E-I
J-L

22 23 24 21
28 27 26 25

2 B P
2066
2065
2020

2189
~~2189~~
2051
2050
35A
HYPER-TEXT-SEARCH → 30000
HYPER-INSTALL
L\$158
L\$189
L\$51
2000

LMA RAM RAND 2 } DK
 LMA RAM RAND 2 }
 [~~RAM RAND 2~~ NO SRAM THERE!]
 LMA RAM RAND 1 } ROM'S
 LMA RAM RAND 1 } ~~RAM~~ } = X
 - [ROM STUFF]
 6F b865 07

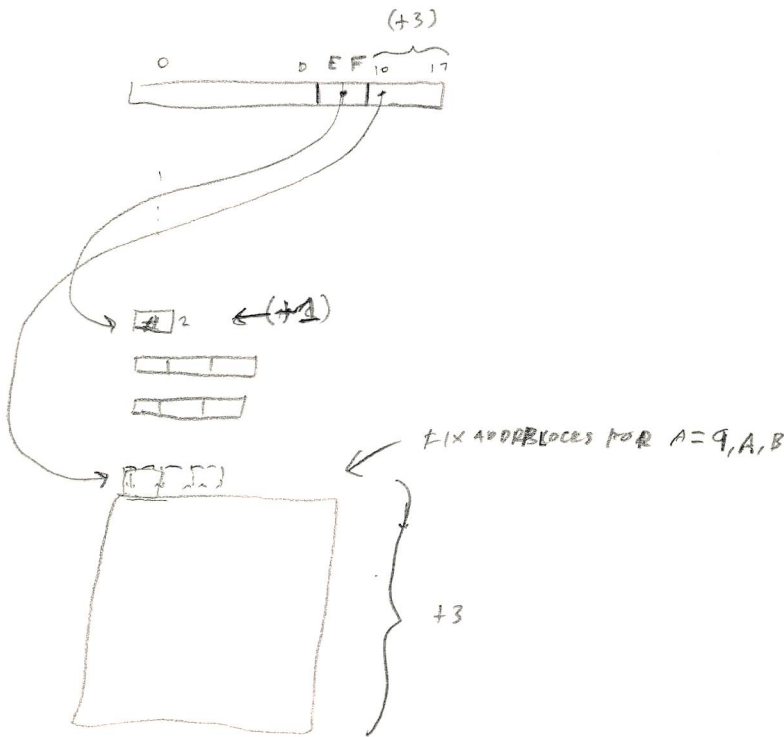
FL 76 L

RAM RAM D1
RAM RAM D2
RAM RAM D1
RAM RAM D2
RAM RAM D1

esc from inhwt

→ FC: E2 E5
FD: F1 A4
FETM: F0 F2

addtent



- ✓ 1. move up block by 3 bytes
- ✓ 2. inc # of bytes
- ✓ 3. get defs + \$12, 11 and store new wt
4. ~~inc~~ inc addr in headlist by 3 (defs + ~~\$12~~ 12) 12
5. fix addr blocks (+3) (defs + ~~\$12~~ 12) 12

inc add
i: (add) → start on block (len)
addr 2 (inc val)

addtent 2EF4
invt 2F84
1: 85b, -2, -3
E2-E4

Wrong address
coming up back in

2FBD is 47

should be: 2
is actually: 9F + 2

f13ice (2)

00010111
00101110
01011100

ihwte

i: b3, y
o: a3to, -2, -3

dest: htab

a3to a3to2 a3to3

$y \leq 71$

sty htab

print mess

ldz #0

sta a3to

sta a3to2

sta a3to3

ldz htab

sr dsp 2p4

sr waitkey

cmp #0 ; <return>

bcc rts

cmp #FF ; <delete>

bcc

cmp #"

plb

cmp # "a" + 1

bge

asl

asl

asl

asl

; forget high nibble

~~...~~

pmc slast

pmc slast

pmc slast

pmc slast

? pf1s1

? invs1

PEO:

flaky
Kvers
ruit
m
b: thap
machid

addtent

16p

read 60d
clear 60d
paper
paper
et/entry

562

646

slast

: a3to a3to2 a3to3

i: , a

1 bits

mzc

asl

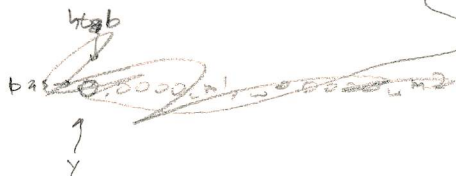
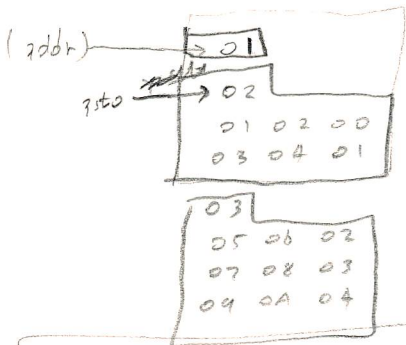
rol a3to3

rol a3to2

rol a3to

com

X cur
X org



~~12x = 8x + 3x~~

save
mult 2
save 2
mult 4
add 2
add 1

$$11x = 8x + 3x$$

in wt
in dm
in comp
in inser

in wt
00.0000
↑ ↑ ↑
asto -2 -3

in dm
00.00 mm
↑
asto -2

dsp 2p2

display 2 dgs "." 2 dgs
: in asto, asto2 bed, b33, y
: as "aa.bb" where aa = asto, bb = asto2
y inc, b3, pres

lda asto
jsr out
lda asto2
jsr out
rts

jsr nibout
fall through

dsp 2p4

jsr dsp2p2
lda asto3
jsr out
rts

dsp 1p4

lda asto
jsr nibout
lda asto2
jsr out
lda asto3
jsr out
rts

EE2

VIAE EB
BASE 20EF

E2 E3 E4
00 00 00

1

has 457 labels

prfld

Q = 9 or 19

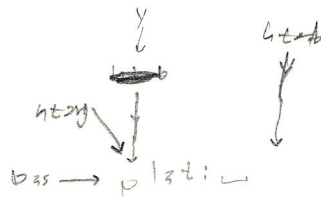
00081001

where $\bar{x}=0 \leftrightarrow$ Vertical
 $\bar{x}=1 \leftrightarrow$ Horizontal

~~as base high~~
~~add~~

ladd: pf9-

X, base Y



asl
 asl
 asl
 pha

; remember (in bit 7) if var H

a = x1001000

lda #9
 sty htab
 sty htaborg
 jsr getaddr

ldy #0
 lda (addr),y

iny
 cmp #0
 beq

sty asto
 ldy htab
 cmp #1

beq
 jsr prncss
 asc "plat: "
 dcb #

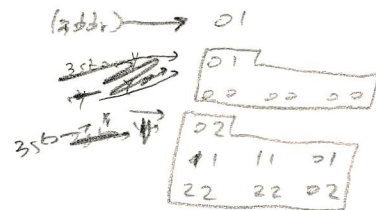
jmp
 jsr prncss
 asc "clad: "
 dcb #
 sty htab
 cpy #80

beq
 ldy asto
 pla
 pla bpl ; (vert)
 jsr prncmp

a = 1

sk1

pla ; Var H in bit 7
 jsr prncmp
 rts



jsr prncmp
 sty asto
 ldy htaborg
 jsr prncss
 asc "core: "
 dcb #
 sty htab
 cpy #80
 beq

sk6

ldy asto
 pla
 pla
 rts

sk5

pla
 rts

; (horiz)

jsr prncmp
 sty asto
 ldy htab
 cpy #80

beq
 jsr prncss
 asc "; core: "
 dcb #

lda #20
 sta vtab
 jsr base/c
 ldy #0
 lda #0
 lda #819
 jsr prfld
 brk

prfunt

i: addr, y, bas, htab
 no-pres: addr, bas
 inc: y, htab
 dest: asto, x

htab
 1.0000 metal 1

ihv
 lda (addr), y
 pha
 dey
 ora (addr), y
 bne
 lda #"
 bne
 lda #"
 sty asto
 ldy htab
 jsr cont
 sty htab
 cpy #80
 beq
 pla
 lda #"
 jsr cont
 sty htab
 cpy #80
 beq
 ldy asto
 lda (addr), y
 iny
 iny
 sty asto
 ldy htab
 jsr cont
 sty htab
 pla
 cpy #80
 beq
 jsr cont
 sty htab
 cpy #80
 beq
 lda #80
 jsr cont
 sty htab
 cpy #80
 beq

pla
 ldy asto
 iny
 iny
 sty

ldy asto
 iny
 lts

prcomp

i: addr, y, bas, htab, a (x.....)
 o: horiz: pres: addr bas vtab
 inc: y htab
 vert: pres: addr htab
 inc: y bas vtab
 dest: asto, asto2, htaborg, x

0 = V
 1 = H

htaborg
 asto2
 03
 01 02 00
 03 04 01
 05 06 02
 0.0102 metal 0
 0.0304 metal 1
 0.0506 metal 2
 bas
 0.0102 metal 0
 0.0304 metal 1
 0.0506 metal 2

21F6 → htab E5
 21EF → vtab E0
 21FC → A
 21F3 → addr F2.F3

00
 00
 2120000000
 51

after 1st loop y not right
 y
 end (y not right)

251

~~251~~

lda htab

sta htaborg

lda (addr).y

sta asto2

iny

bcs

jsr prfamt

~~jsr prfamt~~

~~lda htab~~

~~cpy #0~~

~~bca~~

plp

phs

bcs ; (or if horiz)

;(vert):

~~lda htab~~

~~cpy #0~~

~~bca~~

~~jsr prfamt~~

~~lda htaborg~~

~~sta htab~~

~~inc vtab~~

~~jsr bascale~~

~~jsr prfamt~~

~~sta asto2~~

~~cmp #1~~

~~bca~~

~~lda htab~~

~~cpy #0~~

~~bca~~

~~lda #4~~

~~jsr cont~~

~~sta htab~~

~~cpy #0~~

~~bca~~

~~lda #5~~

~~jsr cont~~

~~sta htab~~

~~cpy #0~~

~~bca~~

~~lda asto~~

~~sta asto~~

jsr prfamt

dec asto2

bne ptele

~~lda htab~~

~~cpy #0~~

~~bca~~

~~lda #4~~

~~jsr cont~~

~~sta htab~~

~~cpy #0~~

~~bca~~

~~lda #5~~

~~jsr cont~~

~~sta htab~~

~~cpy #0~~

~~bca~~

~~lda asto~~

~~sta asto~~

~~jsr~~
bcs

c=x
m

~~jsr~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

~~sta~~

~~cpy~~

~~bca~~

~~lda~~

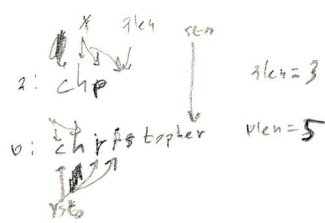
~~sta~~

~~cpy~~

~~in:~~

a, b, aLen, bLen

len 2 aLen



equal: 1

a \$200

b \$300

aLen FE

bLen FF

equal: FD

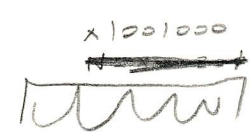
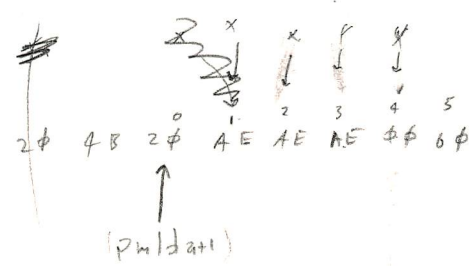
\$4000

q = 9 or 17

000x1001

x1001000

when x = 0 \leftrightarrow Vertical
x = 1 \leftrightarrow Horizontal



(addr) \rightarrow 01
3.5 \rightarrow 0.2
11 22 33
44 55 66
3.5 \rightarrow 0.2
10 20 30
40 50 60

AE A=0



2447

forvent
SDSD print of below line
when anything before + inclg
0.0000x
runs off
for halt:
screen up

17. 3
in buffer

butr: C H R u
C3 C8 b2 pf

ATTN: PERSONNEL
26 FEDERAL PLAZA
NY NY 10278

6083. 60FF

713A
608C

Change: 1

don't inc

blocks: $z = 7.8$

 ~~$(y = E, E; 10, 11)$~~

Ex: $\begin{array}{cccccc} 68 & 68 & AA & 38 & 68 \\ 68 & AA & 68 & 18 & 68 \end{array}$

61 F8; F2 $\phi\phi$ 145

1265, 1267

1296, 1303

vertical
scroll up n cycles

sta page 2
sec
ldy #37

4
2

→ [Ida sea pairs]

X

dey

2

bye

$\uparrow 3, 2 \downarrow$

12

4

place

43

sta pape!

4

- bmi; cs

 $\uparrow L_2$

lets

Handwritten notes illustrating the execution of a recursive function for calculating Fibonacci(4):

Left side (Call Stack):

- 4 (initial call)
- 2 (returning 2)
- 2 (returning 3)
- 4 (returning 5)
- 2 (returning 8)
- 4 (returning 13)

Right side (Execution Flow):

- 2 (initial call)
- 2 (returning 2)
- 3 (returning 3)
- 4 (returning 5)
- 3 (returning 8)
- 4 (returning 13)

Bottom:

HTS 28

$$2563 = \text{constant}$$

ben x

X on

do. 4.

for $a = 1, 2, 4, 5, 6, 4$:


$$\rightarrow (A(2B)) \div 3 + len \quad \checkmark$$
$$\text{addr} + 8 + \text{len} \rightarrow \text{addr} + 3$$

1da addr
 adc #0 f add 1
 sta addr
 lda addr+1
 adc #0 : clear C
 sta addr+1 jmp
 → ~~1da~~ (addr), y
 adc #2
 sta (addr), y
 iny
 lda (addr), y
 adc #0 : clear C
 sta (addr), y
 lda addr
 adc #2
 sta addr
 lda addr+1 : clear C
 adc #0
 sta addr+1
 dex
 bne addr

21

```

d, # 0
lda addr3
sta (addr), y
iny
lda addr3 + 1
sta (addr), y

```

previously moved

update

1. display current definitions ~~[~~by~~ ~~subject~~]~~ ^{obj.}
2. ~~use~~ ~~Arrows~~ to move thru defs.; ↑: \$88 or \$8B ↓: \$8A or \$95
 <return> to edit a def.; \$8D
 A to add a def.; \$C1
 <esc> to quit. \$9B

<return>:
edit a def.

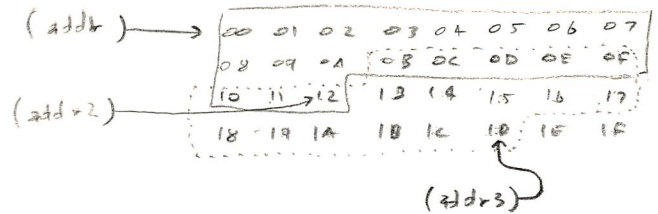
<arrow>'s move through fields
<return> to work on a field
~~A to add a new one to the current list~~
~~<esc> to quit~~

<return>:
work on a field

<arrow>'s to move thru current choices
A to add a new choice

addr - addr2
 move back up \$B bytes

\$264D
 move up B



```

clc
lda addr2
adc #B
sta addr3
lda addr2+1
adc #0
sta addr3+1
  
```

addr3 := addr2 + \$B

addr: 01 04
 addr2: 02 04 0F 04
 addr3: 01 20
 FE 1F

\$264D
 move up

```

ldy #0
lda (addr2), y
sta (addr3), y
  
```

~~ldx addr2~~
~~cpx addr~~
~~bcc~~
~~dec~~
~~cpx~~

```

ldx addr2
bne
dec addr2+1
cpx addr
bcc
  
```

0402 → 2001 ✓
 0401 → 2000 ✓
 0400 → 1FFF ✓
 2601 → 1FFF ✓

```

lda addr2+1
cmp addr+1
bcc cpx+1
bne
  
```

```

lda addr3
bne
dec addr3+1
dec addr3
bne (bra)
  
```

```

lda addr3+1
cpx #4
bne
adc #0 (add 1)
cmp addr+1
bne
  
```

addr \$F2-F3
 addr2 \$F4-F5
 addr3 \$FC-FD

r: FE 7F
 r2: FE 7F
 r3: 00 80

A=11

X=FF

7FFF → 8001 ✓
 7FFE → 8000
 7FFD → 7FFF
 7FFC → 7FFE

	3B	55	88	20	40	71	7 A2	8 A9	9 AE	A CE	B 04 62
18 60	30 60	4A 60	7D 60	85 60	42 61	66 61	97 61	9E 61	A3 61	C3 61	F9 61

6018: 02 00

601A: 0 0 0 0 1 1 0 0 0 0 1

6025: 1 2 2 1 3 3 1 0 1 1 2

6030: 2 35 60 3A 60

6035: E3 E5 EE F4 00

603A: F6 ... E5 00

604A: 4 53 60 5B 60 68 60 72 60

hex: F8, F9

coin: FA, FB

coin: F6, F7

~~2957 Ad~~

145

175 ← 2768

 $x, y, z, 2x+2, 2y+3$

Index of data blocks
~~add of~~ with address to
 fix (+3B)

incabB

i: addr

ldy #0

ldx (addr), y ; # of addresses

asl

sty asto

inc asto

; first offset
; first ~~addr~~ 1st used

ldy #1

~~ldx #0~~

clc

lda (addr), y

adc #B

sta (addr), y

iny

lda (addr), y

adc #B

sta (addr), y

iny

cpy asto

bne

rts

(addr) → 05, 0

, 1, 2

00 01

, 3, 4

F.F 02

, 5, 6

80 38

, 7, 8

61 52

, 9, 10

F.F 00

, 11, 12

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

18

hex values

(same)

0B 01

0A 03

8B 38

6C 52

09 01

incabB: \$26D4

i: F2-F3

(addr) → 18 60

incabB: \$26E5

i: F2-F3

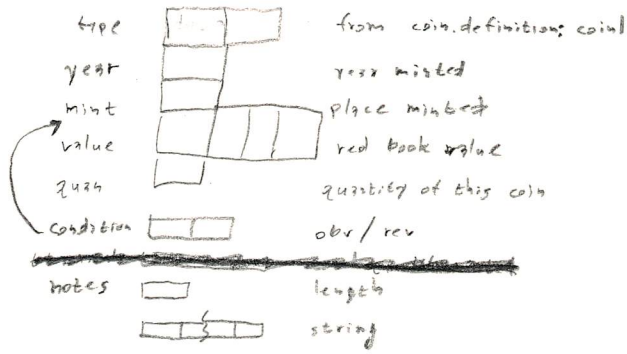
(addr) →

~~incabB~~

COIN

coin data

each coin:



bytes

2
1
1
4
1
2
1
x } 0 ≤ x ≤ 255

print comp list

i: hlab, bas, addr, y,
a (#mclab)

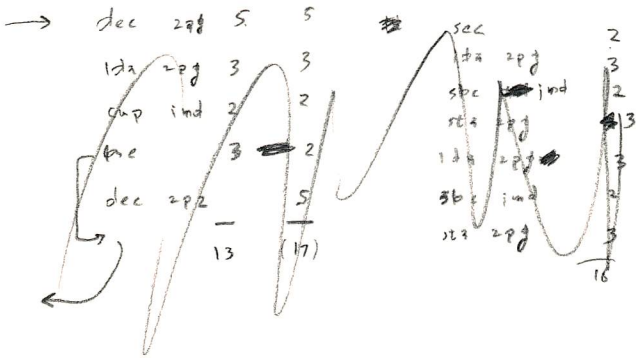
23D5:

print mcl1

i: 2(mcl1#), bas, asto3(hlab)
o:

242E:

lay
lda #A
jsr psetaddr
ldy asto3
jsr prstr
rts



idx coinsum 3x
dex 2x
cpx HIFI 2x
btx coinsum 3
bne 3
dec coinsum 1

21E4

addr: F2.F3

addr2: F6.F5

huncos:

02 40

coinsum:

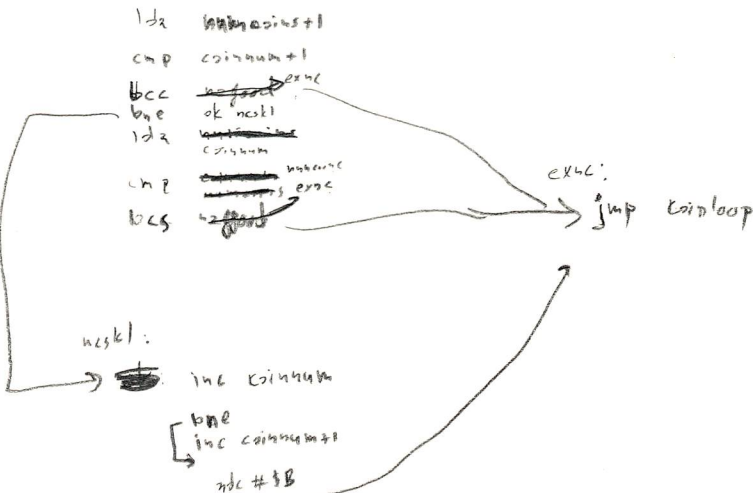
FF FF 00 01 03 02 40

huncos

coin sum 01 40 ... 00 00

valid

ok



COIN

files

prodes	SYS	Prodes system	
coin.system	SYS	main rxu/vcl/view system (Δ coin.data)	A\$2000
coin.update	SYS	update utility (Δ coin.defn)	A\$2000
coin.data	BLN	list of coins	A\$2000 (after coin.system)
coin.definition	BLN	coin/year/m data	A\$2000

display a coin definition

1 OTTER

cent 2

4 Abraham Lincoln

5 "IN GOD WE TRUST LIBERTY (date) (mm) "

6 by Victor D. Brenner

7 edge: "plain"

8 wheat stalks

9 "E. PLURIBUS UNUM ONE CENT UNITED STATES OF AMERICA"

10 by Victor D. Brenner

weight 3.11 g

diameter 19 mm

composition .9500 copper, .0500 tin and zinc

20
21
22
23

addr → 0

2
00 75 00

01 25 00

1
00 00 00

inc \$t4
lsh (388), Y
st4 (388), Y
16

inc \$t4
lsh (388), Y
st4 (388), Y
4
20

addr → 0 3100

2
00 00 00
01 34 56
4502

Y → 01

mul2
lsh 4 FFFF
st4 5 FFF
inc mul2+1
bne mul2+2
inc mul2+1
bne mul2+2
inc mul2+2

350 = 00 (metal)

13
hlt
1.0000
copper
3503

3 = 28 00

2
01 34 56
56

3 = 2

1 0

3 = 0, 1
350 metal

350 = 01 (metal)

1
1

13
hlt
1.0000
copper
3503

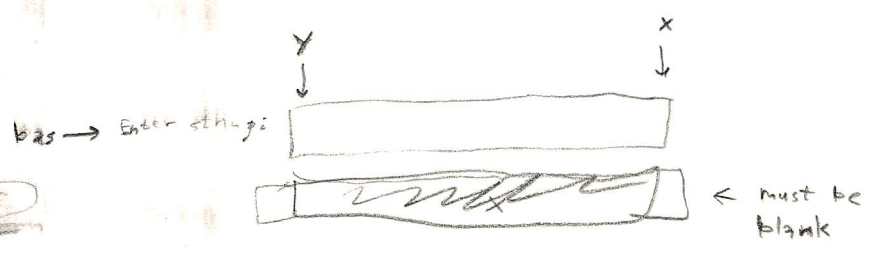
1.0000
copper
3503

upmove
have look up 38 bytes

5 start
5
3502
3503

~~13~~
~~hlt~~
~~1.0000~~
~~copper~~
~~3503~~

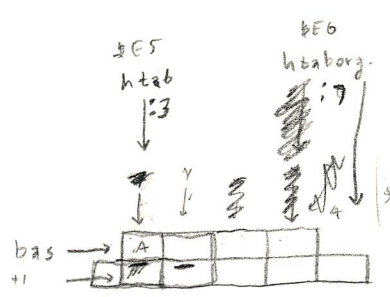
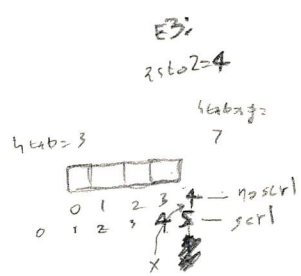
instr
 vtab ← (scrn pos) ≤ 22
 X (last col to display) ≤ 78
 or: htab = string (ASCII \$A0-\$FE)
 (term w/ 00; no \$80)



key send functions

- ASCII:
- \$A0-\$FE (00) ~ enter character
 - \$88 (left-arrow) X sign
 - \$95 (right-arrow) back-nuclear
 - \$FF (delete) back-clear
 - \$9B (esc) quit = htab empty
 - \$8D (return) done
 - ~~\$80 (tab) insert~~

B L T
 0 2 00 14
 0000 0000 0000
 1 3 2 8



1-72
 if X (73+) then scroll it

scrinlt
 scrinrt
 i: htab, htaborg
 or: pres: bas, X, Z

input: 0, 1, 2, 3, 4, 5, 6
 xsto (len)
 \$E2

1336-1407

if Y is even
 in Aux X ← M
 if Y is odd
 in Main M ← X
 if Y is even
 in Aux M → X
 if Y is odd
 in Main X → M

tya
 pha
 tsv
 tsv
 bcl
 sta page2
 iny
 lda (big), Y
 sta page1
 dey
 sta (bas), Y
 pla
 tay
 rts

fld menu

i: 2 (as prfld: ~~1-A~~)

X (which one to start with)

0: X (which one chosen)

0		
1		
2		
3	3	x-2
→ 4	entry 0	x-7
5	6	entry 1
6	5	entry 2
7	4	entry 3
8	3	entry 4
9	2	entry 5
10	0 1 2	entry 6
→ 11	→ 0	entry 7 *
12		entry 8
13		entry 9
14		entry 10
15		entry 11
16		entry 12
17		entry 13
18		entry 14
→ 19		entry F x+8
20		<u>size 10</u> x+9
21		
22		
23		

prfld: i: bas, y
2, x

dest: add, y, 4to6, x, x, xsto, add, 3, xto2

START

print "-7"

if x > 7 then VAB4

print entry # X-7

→ VAB 17 or last entry

~~the first entry~~
~~the last entry~~

292

-29=3

if x < 7 then

VAB 11-X

print entry #

→ VAB 17 or last entry

copy 13, 145 to 1159?

jsr esc after prfld

←

A9 04	VDA #504
A2 09	LDX #509
20 6E 20	JSR \$296E
00	BRK

2528: prfld

```
ldz #18
sta vtab
jsr bascalc
ldy #4
lda (con), y
tax
lda #9
ldy #13
jsr prfld
```

print individual field

1: bas, y (vtab, htab)

a: (con)

- 0 den
- 1 sub
- 2 insc
- 3 inschr
- 4 desg
- 5 edge
- 6 weight 17% #
- 7
- 8 diameter 18% mm
- 9 comp 19 cone ←
- A metal

X: which one in list ← don't check for validity

first: addr, htab

Y = htab for next print

dest

1, 2, 4, 5, A: addr, htab := Y

3 addr, htab

6 addr, htab

7 addr, htab, addr2

17

18

19 ← addr, addr, htab
blackout, auto, 2, 3
addr2

for a=1, 2, 5, 6, A:

lbr htab

```
asl
tay
clc
lda defs, y
adc #1
sta addr
iny
lda defs, y
adc #0
sta addr+1
lxa
asl
tay
lda (addr), y
tax
iny
lda (addr), y
sta addr+1
stx addr
```

453, 471

EXAMP: lbr #0
lbr #1, y

which one

580 - 443

```
ldz #2
sta vtab
jsr bascalc
lda #1
ldy #5
jsr prfld
```

ldy #0
ldy (con), y
tax

2515
1
322

465

423 - 450

dest

prfld
#2, X
(as prfld)

ldy htab
jsr prfld

addr
y
already
addr, bas, y

prfld

546

548

ldz #3A2

jsr count

lbr htab
lbr for addr2
ldy htab
CPY

jsr prfld

bcc

ldy #76

jumpress
to "..."
WFB
ldy #79

ldz #3A2

jsr count

rts

ACK
STEPH
AT 8:10

print composition (1=9)

X = which one X

bas = base addr for print

$y = \text{label}$ for print $x(\text{label})$

$X \xrightarrow{\text{getaddr2 (addr)}} 1$
 $X \xrightarrow{\text{getaddr2 (addr)}} 2$
 print
 int x (4 bytes)
 25 00 00
 75 00 01
 3
 10 00 02
 30 00 03
 50 00 04

was ~~100~~ ct ad: x X. 2500 copper
x. 7500 nickel
→ ebe: x X. 1000 steel
X. 3000 zinc
X. 6000 silver

$$(n=17)$$

$\therefore z = 19$

$x = \text{which one}$

bas: base

$$\gamma = 4\pi ab$$

$x \left\{ \begin{array}{l} \text{potato} \\ \text{add} \end{array} \right. \rightarrow 1$
 $x \left\{ \begin{array}{l} \text{potato} \\ \text{add} \end{array} \right. \rightarrow 2$
 $x \left\{ \begin{array}{l} \text{potato} \\ \text{add} \end{array} \right. \rightarrow 250000$
 $x \left\{ \begin{array}{l} \text{potato} \\ \text{add} \end{array} \right. \rightarrow 250000$
 $x \left\{ \begin{array}{l} \text{potato} \\ \text{add} \end{array} \right. \rightarrow 3$
 $x \left\{ \begin{array}{l} \text{potato} \\ \text{add} \end{array} \right. \rightarrow 100002$
 $x \left\{ \begin{array}{l} \text{potato} \\ \text{add} \end{array} \right. \rightarrow 300003$
 $x \left\{ \begin{array}{l} \text{potato} \\ \text{add} \end{array} \right. \rightarrow 100004$

Q25 \rightarrow class x.2500 copper

print weight (a=7) aa block

$$1: 2 = 7$$

$x =$ which one

bag = base

$$Y = 4tab$$
$$(defsf \rightarrow 04$$

0:	1	00	00	00
1:	4	11	11	11
2:	7	22	22	22
3:	A	33	33	33

 $D \rightarrow$

~~1-358~~

~~3-731~~

OFF1

M 7.CS6 1

F 7.CS6 2

11-7431

~~C.S.5~~ 5

D T.CU1 2

C.D.5 —

~~1904CS~~

REL 6 TRS
MATH 7
EQU 1
START 3 2
MATH 2

$$\begin{array}{r} X * 3 \\ + \\ defg + E \\ + \\ 1 \end{array}$$

276 - 318
284 - 314

REMAINS

✓ T. PDE ERU	PDE.1
FADE DEL	PDE.1
✓ T. PDE START	PDE.1
✓ T. PDE DATA	PDE.2
PDEMAIN.S	PDE.2

025msd A030-AD67

$x = 80$

32 ~~44~~ 0

75 to 280

PORE L-5

1

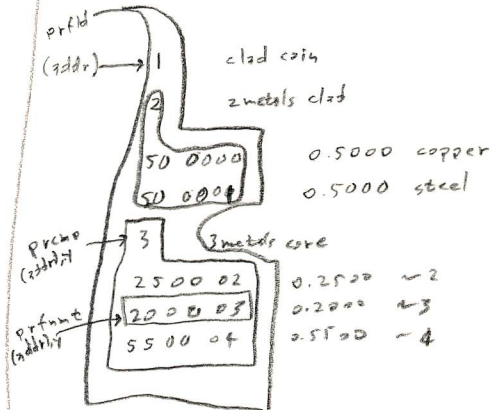
1420

2100-3FFF
↓
4000-BEFF

2000 cu start

2050 start to stop PRODS stuff

composition list



0 1 2 3 4 5 6 7 8 9 A

ecvt: ~~11~~ 2 5 6 7 A D E F 11 14 11

echt: ~~11~~ 0 8 8 8 8 8 8 7 9 17

ecchr: 0 0 0 0 0 0 0 80 80 80

PL M1

prfld

i: a, x

bas, y

→ 1 clas

2

50 00 00

50 00 00

3

25 00 02

20 00 03

55 00 04

prfms print fineness and metal

i: (addr), y → KK FF mm

dest: ~~2500~~, x

bas, htab

o: pres: addr, bas inc: y, htab

prcms print one list

i: (addr), y → # metals

bas, htab

z: 80 = horiz.

00 = vert.

(vert. htab)

o: pres: addr, ~~bas(horiz), htab(horiz)~~ (horiz: ~~bas~~, bas, htab)

inc: y, ~~htab~~, ~~bas(vert)~~, ~~htab(vert)~~ (vert: ~~bas~~, bas, htab)

(horiz: htab)

dest: ~~2500~~, ~~2500~~

htaborp, x

2557:

~~2605: prcomp~~

~~2605: prcomp 2~~

pr fid → 242C

2571:

asl
asl
asl
pha
sty htab
jsr getaddr2
ldy #0
lda (addr), y ; ahan, clat, plat

(addr) → 1

0 0 0
y → 2
0 0 0
1 1 1

pl:
09 or 19

bas } scripps
htab }

iny
pla
h 100 1000
jsr prcomp 2
rts

bas → clad: 1.0000 copper, 1.0000 zinc; clad: 1.0000 copper

cmp #1

brc ; br if plat

jsr prcomp
asc "clad: "
dfb d

(bra)
jsr prcomp
asc "plat: "
dfb d

sty htab
ldy asto

pla
pha
jsr prcomp 2
sty asto
pla
pha
bpl ; br if vert

ldy htab
lda #1
jsr cout
lda #14
jsr cout
sty htab
bne

sec
lda htab (\$13)
sbc #6 (\$10)

jsr prcomp
asc "core: "
dfb d
sty htab
ldy asto

pla
jsr prcomp 2
rts

ES: htab

F2.F3: addr
A Y

hopt 21E3
cart 21FD

prid 2-9, 17

22

3E:

24: lda #9

ldy #13 \$00

jsr prfid

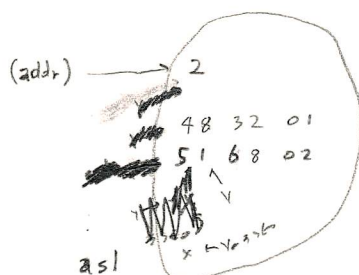
rts

2427

pr cmp

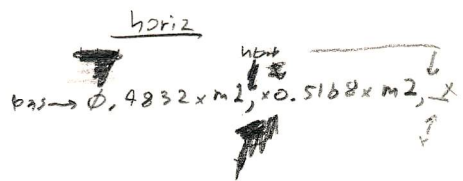
63
core: 1.0000 m...

2 = 84



htaborg vert

0.4832 x metal 1
0.5168 x metal 2
htab
x



asl
php
lda htab
sta htaborg
lda (addr), x

; # of metals (20)

astob ~~phi~~

dest: asto2
htaborg
asto
x

iny
sta asto2

; # of metals (>0)

C=1,0(b,u)

jsr prfmt
sty asto
plp
php

; br if horiz

lda htaborg
sta htab
inc vtab
jsr bascale

bcc (bra)

lda asto2
cmp #1
bca
lda htab
lda #"
jsr cout
lda #5AF
jsr cout
sta htab
lda asto

dec asto2

bne

pla

rts

cpy #80
beq

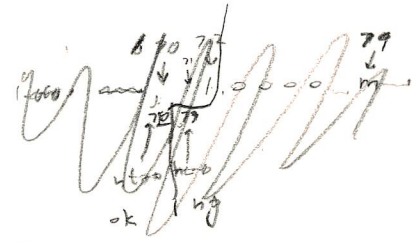
lda #12
sta vtab
jsr bascale
lda #20
sta htab

lda #5AF
sta addr
lda #61
sta addr+1
ldy #0
lda #0

jsr prcomp2 268E
rts

vtab: E0: 12
(in)bas: F0: 140
htab: E5: 20
addr: F2: AF 61

61F3, 61DF



check:

of htab

cpy #70
bcc

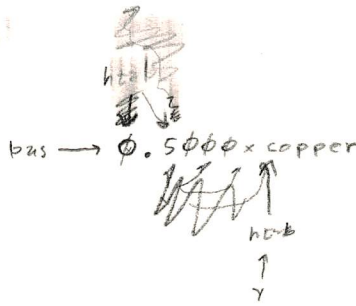
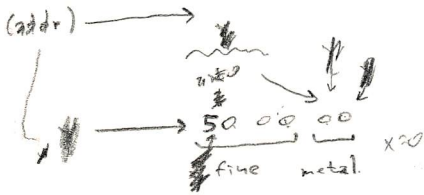
jsr prcomp
sta htab
lda asto
pla
bne

cpy #78
bcc

lda #77

prfmt

~~prfmt~~



addr = F2.F3

bas =

hnb =

iny
lda (addr),y
pha
ora (addr),y
bne

lda #1

bne

lda #0

sty asto

ldy htab

jsr cout

lda #1

jsr cout

sty htab

~~lda asto~~

~~lda (addr),y~~

ldy asto

lda (addr),y

iny

iny

sty asto

ldy htab

jsr cout

pla

jsr cout

lda #AF

jsr cout

sty htab

ldy asto

lda (addr),y

tax

lda #A

ldy htab

jsr prfld

~~sty htab~~

ldy asto

iny

rts

test: asto, x, y

61AF: 02

61B0: 01

61B1: 00 00 03

61B4: 01

61B5: 00 00 02

(map) F5 = 20

(asto) E2 = 2

21E3: lda #12

sta vtab

jsr bascale

lda #AF

sta addr

lda #61

sta addr+1

lda #20

sta htab

ldy #2

jsr prfmt

rts

F2: AF 61

Y = 2

4572

coin definition (p. 1)

\$66666

list of addrs of:

coin:	#:	0
den:		2
inset		
inset		
desig		
weight	edge	10
diam	spec	12
comp		14
comp		16
comp		18

coin:

coin types: 2 bytes

02 00

list of coin types:

00 00 00 00 01 01 01 00 00

each coin type is 4 bytes:

0 000 00 00 01 01 02 00 01

den, ~~side~~ side, edge, spec

00 05 01 01 01 01 01 01 01 01

obv:

obs: 1 byte

01

list of obs:

each obs is 3 bytes:

00 00 00

sub, inset, desig

rev:

revs: 1 byte

02

list of revs:

each rev is 3 bytes:

00 01 01

sub, inset, desig

02 01 01

edge:

addrs: 1 byte

03

list of addrs of edges in order up from 0

edge0 edge1 edge2

list of edges:

ca: string #

0: [plain] #

1: [creeded] #

2: two HUNDRED FOR A DOLLAR #

spec:

specs: 1 byte

02

list of specs:

each spec is 3 bytes:

00 00 00

wt, dia, comp

00 00 01

den:

den: 1 byte

02

list of addrs:

den 0 den 1

list of den's:

0: balanced

ca: string #

1: cent #

inscl:

inscs: 1 byte

6448: 02

list of addrs of inscs in order up from 0

inscl inscl

list of inscs.

ex: km; inscstring

6445: 0: d3

03 A0 00

6449: 1: 06

04 AA 03 04 A0 00

inschr:

chars: 1 byte

list of addrs of chars in order up from 0

list of chars;

ex: string ap

6450: 06

addrs 0, 1, 2, 3, 4, 5

(3 dummies)

6450: 0:

~~Edwards ap~~

1: ~~LIBERTY~~

2: E PLURIBUS UNUM 00

3: LIBERTY 00

4: ~~LIBERTY~~ 00

5: IN GOD WE TRUST 00

desgl:

dgs's: 1 byte

02

list of addrs of dgs's in order up from 0

desgl desgl

list of dgs's;

ex: string 00

0: Adam Eckfeldt 00

1: Robert Scot 00

weighth:

weights: 2 byte

02

list of weights;

ex: 3 bytes ^(gibcc) in BCD:

aa. bbcc g

6088:

addr → 22 00 00

21 50 00

(22.0000 g
21.5000 g)

diam:

diam's: 1 byte

01

list of diam's;

ex: 2 bytes (aabb) in BCD:

aa. bb mm

addr → 11 45

(11.45 mm)

subj:

comp's: 1 byte

1st of addrs of comps is order up from

list of coms: c2:

[illegible]

02

comp0 comp1 comp2

01	20 00 00	Capex	1.0000
02	00 88 00	Capex	.8800
	01 12 00	nickel	.1200

compn!

metals: 1 byte

lost of addrs of metals is order up from P

lost of metals:

ca: string 00

04

Exoph 0, comp 1, 2, 3

0: correct FF

346-3 1: nickel pp
 2: steel pp
 3: zinc pp

clad, plat?: 1 byte (0=home, 1=clad, 2=plat)

if 0: 1 metal list

rf 1,2: 2 metal lists

metal lists:

metals: 1 byte

list of: ~~_____~~

fineness: 2 bytes BCD:

2266

 $(0,0 = 1.0000)$

metg1: 1 byte

0 0 1 0 0 0

1:  2

$$\frac{1}{A} \rightarrow 2, 3 \text{ to } 2$$

2/11/21

$y \rightarrow 007500$

10

01 25 30

1

11. 11. 11

000079

11/11/11

2. 11/11/11

2. 2 VV

1871

03 0p 00

1

02 00 00

$$\frac{A}{u}$$

4

obverses

?

subject:

	<u>position</u>	<u>facing</u>
Liberty →	head →	left
Indian	seated	right
Lincoln	standing	
Jefferson	walking	
Roosevelt		
Washington		
Franklin		
Kennedy		
Eisenhower		
Anthony		
eagle		
shield		
star		

designer:

list of designers

coin. definition

coin:
 detail
 obverse
 reverse
 edge
 specifications

obverse:
 subject
 inscription
 designer

reverse:
 subject
 inscription
 designer

specifications:
 weight
 composition
 diameter

edge:
 { plain
 reeded
 lettered
 otherwise }
 → string
 → string

subject:

designer:

inscription:
²
 (2 types of ins.)
 string w/ ins.chars, ⁰date + mm will be added
 during use (except poss. "P")

weight:
 3 bytes BCD:
 xx.yy zz g

diameter:
 2 bytes BCD:
 xx.yy mm

ins.chars:
 3 LIBERTY
 7 UNITED STATES OF AMERICA
 5 IN GOD WE TRUST
 6 E PLURIBUS UNUM
 4 * + + + + * (6 stars)
 5 IN GOD WE TRUST
 0 IN GOD WE TRUST.
 ...

composition:
~~fineness~~ fineness - metal

0	Lhl cap&pole	Adam Eckfeldt?	LIBERTY 1793
1	Lhr cap&pole	Robert Scot	LIBERTY 1794
2	Lhr cap&pole	John Smith Gardner	LIBERTY 1795
3	Lhr draped bust, stars	Scot	LIBERTY ***** 1796
4	Lhl classic	John Reich	LIBERTY ***** 1809
5	Lhl (braids) coronet	Christial Gobrecht	LIBERTY ***** 1840
6	Lhr flowing hair	Henry Voigt	LIBERTY 1793
7	Lhr flowing hair	Eckfeldt	LIBERTY 1793
8	Lhr cap&pole	Joseph Wright	LIBERTY 1793
9	Lhl coronet	Scot	LIBERTY ***** 1816
10	Lhl coronet	Gobrecht	
11	eagle flying	James B. Longacre	UNITED STATES OF AMERICA 1856
12	Indian h	Longacre	LIBERTY UNITED STATES OF AMERICA 1859
13	Lincoln h	Victor D. Brenner	IN GOD WE TRUST LIBERTY 1909
14	shield	Longacre	IN GOD WE TRUST 1864
15	star	Longacre	UNITED STATES OF AMERICA 1851
16	Lhl coronet	Longacre	LIBERTY UNITED STATES OF AMERICA 1865
17	Lhl coronet	Charles E. Barber	LIBERTY ***** 1883
18	Indian h	James Earle Fraser	LIBERTY 1913
19	Jefferson h	Felix Schlag	IN GOD WE TRUST LIBERTY 1938
20	Lhr flowing hair	Scot	LIBERTY ***** 1794
21, 22	Lhl capped classic (2)	Wm. Kneass	LIBERTY ***** 1837 UNITED STATES OF AMERICA (1837) ← 2
23	L seated	Gobrecht	LIBERTY ***** 1837 IN GOD WE TRUST (1837) ← 2
24	Lhr wreath	Barber, C.	LIBERTY UNITED STATES OF AMERICA 1892
25	Lhl winged	Adolph A. Weinman	LIBERTY IN GOD WE TRUST 1916
26	Roosevelt h	John R. Sinnock	LIBERTY IN GOD WE TRUST 1946
27, 28	L seated (2)	Wm. Barber	LIBERTY ***** 1875, (+) IN GOD WE TRUST
29	L standing	Herman A. MacNeil	IN GOD WE TRUST ***** LIBERTY 1916
30	Washington h	John Flannahan	IN GOD WE TRUST LIBERTY 1932
31	Lhl classic	Gobrecht	LIBERTY ***** 1836
32	L walking	Weinman	IN GOD WE TRUST LIBERTY 1916
33	Franklin h	Sinnock	LIBERTY IN GOD WE TRUST 1948
34	Kennedy h	Gilroy Roberts	LIBERTY IN GOD WE TRUST 1964
35	Lhl coronet, capped	George T. Morgan	LIBERTY E·PLURIBUS·UNUM ***** 1878
36	Lhl crown (peace)	Anthony DeFrancisci	IN GOD WE TRUST LIBERTY 1921
37	Eisenhower h	Frank Gasparro	IN GOD WE TRUST LIBERTY 1951
38	Susan B. Anthony h	Frank Gasparro	IN GOD WE TRUST LIBERTY 1979
39, 40	Indian h (2)	Longacre	LIBERTY UNITED STATES OF AMERICA LIBERTY ***** (no date!)
41	Lhr capped	Scot	LIBERTY ***** 1796
42	Lhl classic, capped	Reich	LIBERTY ***** 1809
43	Indian h	Bela Lyon Pratt	LIBERTY ***** 1908
44	Lhl curly hair	Barber, C.	LIBERTY ***** 1908
45	Lhl curly hair	Barber, C.	LIBERTY ***** 1908
46	Indian h	Augustus Saint-Gaudens	LIBERTY ***** 1907
47	L walking	Saint-Gaudens	(46's) LIBERTY 1907
48	Lhl classic, capped	Reich	LIBERTY ***** 1813
49	Lhr draped bust	Scot	LIBERTY ***** 1796
50	Lhl coronet	Longacre	LIBERTY ***** (no date!)
50	Lhl coronet	Longacre	LIBERTY ***** 1849

580.00

obvies:

condition: EF-40

5/1/1919

subject: Liberty head facing left with coronet

47

inscribed: "C. 1851" [illegible]

designer: Christian Gobrecht

review:

condition: FF-40

~~XXXXXXXXXXXX~~

subject: wreath

Area

description: "UNITED STATES OF AMERICA ONE CENT"

designers: Christian Gobrecht

edge:

[plain]

weight:

3
10.89

~~422~~

Composition:

10000 Copper

diameter:

(27) 5 in m



notes: "1851" ~~3/4~~ "1831 [inverted]" on obverse.

1255 cent EF-40 ~~100~~ 1 @ \$80.00

1795 half dollar G-4/AG-3 1 @ \$525.00

1970cc dollar F-12 1@ \$225.00

color

~~data structures~~
~~condition~~ ~~year~~ ~~compos~~
cond / year

condition

* t t t s s s s

where

ttt = T
ssss = S

~~SHELDON:~~

S!	0	AG-03
	1	G-04
	2	VG-08
	3	F-12
	4	VF-20
	5	VF-30
	6	EF-40
	7	EF-45
	8	AU-50
	9	AU-55
	10	MS-60
	11	MS-63
	12	MS-65
	13	MS-67
	14	MS-70

offset
in byte
value
table
SHELDON.

TEXTCOND:

T:	0	AG
	1	G
	2	VG
	3	F
	4	VF
	5	EF
	6	AU
	7	MS
	8	Proof

offset
in addr table
of word →
TEXTCOND.

weight

3 bytes (BCD)

min 00.0000
max 99.9999

a a . b b c c (g)

diameter

2 bytes (BCD)

a a . b b (mm)
min 00.00
max 99.99

compos

- 0 copper
- 1 nickel
- 2 tin and zinc
- 3 zinc
- 4 silver
- 5 manganese
- 6 steel
- 7 gold

year (good → 2048)

offset from 1793

offset

bars

vine (silver & leaves)
gapped
needed

edges

edges

1, 2, 3

- 0 plain
- 1 needed
- 2 lettered
- 3 other



~~United States Regular Issues~~

style

type	value (\$)	years minted	# years	71 subjs	obverses
0 half cent	0.005	1793 - 1857	65		0, 1, 2, 4 ⁸ , 4, 5
1 cent	0.01	1793 - date	196 +	x	6, 7, 8, 2, 4 ⁸ , 4, 9, 11, 12, 13
2 two-cent piece	0.02	1864 - 1873	10		14
3 three-cent piece (Ag)	0.03	1851 - 1873	23		15
4 three-cent piece (Ni)	0.03	1865 - 1889	25		16
5 five-cent piece	0.05	1866 - date	123 +	x	14, 17, 18, 19
6 half dime	0.05	1795 - 1873	79		20, 21, 23
7 dime	0.10	1796 - date	193 +	x	23, 42, 23, 24, 25, 26
8 twenty-cent piece	0.20	1875 - 1878	4		27
9 quarter dollar	0.25	1796 - date	193 +	x	3, 42, 23, 24, 29, 30
10 half dollar	0.50	1794 - date	195 +	x	20, 3, 42, 31, 23, 24, 32, 33, 34
11 dollar (Ag)	1.00	1794 - date	195 +	x	20, 3, 23, 28, 35, 36, 37, 38
12 dollar (Au)	1.00	1849 - 1889	41	x	40, 39, 40
13 quarter eagle	2.50	1796 - 1929	134	x	41, 42, 10, 22, 5, 43
14 three-dollar piece	3.00	1854 - 1889	36		40
15	4.00	1854 - 1889	36		40
16 half eagle	5.00	1795 - 1929	135	x	41, 42, 10, 22, 5, 43
17 eagle	10.00	1795 - 1933	139	x	41, 5, 46
18 double eagle	20.00	1849 - 1933	85		50, 47

gethex1

is bas, y, 2 (starting value)
(scrupus)

0: asto
dest: htab

1. print 2 in inverse

2. get value → asto

3. print asto normal

pitch width: 13.2"
paper length: 25.4"

sec
sbc #7

htab
fix bit

edit:

~~A9A4:DD~~

~~A9A5:48~~

A908: cmp #58D

jsr workfid

20 A9 A9

A915: jmp edrefet

A9B7

1101

monthtbl 13th month
typetbl (pp)
typstbl
sttbl

1256, 1295 p.to 3

~~asto~~
0001020304050607
↑
htab

10 11 12 13 14 15 16 17
00 01 02 03 04 05 06 07
↑
asto 2
x=0

	# bits	values	(range)
Year	7	0-127	1900-2027
Month	4	0-15	(0)-12 (13, 14, 15)
Day	5	0-31	0-31
Hour	8	0-255	0-23 (24-255)
Minute	8	0-255	0-59 (60-255)

↓
0 1 2 3 4 5 6 7
asto = 0 0 0 0 0 0 0 0 ← current value
dnb 0 0 0 0 0 0 0 0
asto2 = 0 0 0 1 0 0 0 0 ← which bit were at
asto3 = 0 0 1 0 0 0 0 0
1 1 1
11101111

240

Months

return
displays
month (asto = 0 into wfbait)
(curaddr)

fields and how they're displayed

#	abbr	name	dtype	display	fh#, #
0	st	storage type	6	\$x.. <stor>	(fh#, # → first column printed after a stor.)
1	nl	name length	7	\$x	(high underlines below name)
2	nam	name	8	cccccccccccccccc	
3	trp	type	9	\$xx.. <trp>	
4	kb	key block	1	\$xxxx	
5	bu	blocks used	1	\$xxxx	
6	len	length	2	\$xxxxxx	
7	cre	creation	5	\$xxxx, \$xxxx.. <datetime>	
8	ver	version --> minimum	4	xx.. --> xx	
9	acc	access	10	\$xx.. DMB \$\$\$PWR	
A	aux	auxiliary	1	\$xxxx	
B	mod	last modification	5	\$xxxx, \$xxxx.. <datetime>	931-10610
C	hdb	header block	1	\$xxxx	
D	res	reserved	3	xxxxxxxxxxxxxxxxxx	
E	enl	entry length	0	\$xx	
F	epb	entries per block	0	\$xx	
10	flc	file count	1	\$xxxx	
11	bit	bit map block	1	\$xxxx	
12	bpu	blocks per volume	1	\$xxxx	
13	pbl	parent block	1	\$xxxx	
14	pen	parent entry number	0	\$xx	
15	pel	parent entry length	0	\$xx	

dtype:

4	0	\$xx
8	1	\$xxxx
1	2	\$xxxxxx
1	3	xxxxxxxxxxxxxxxxxx
1	4	xx.. --> xx
2	5	\$xxxx, \$xxxx.. <datetime>
1	6	\$x.. <storetype>
1	7	\$x (n/high underlines)
1	8	cccccccccccccccc
1	9	\$xx.. <type>
1	10	\$xx.. DMB \$\$\$PWR

DATE

date +1

date

$Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, m_0$

$m_1, m_2, m_3, d_0, d_1, d_2, d_3, d_4$

lstr date

$\phi Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6$

m_0

lstr date+1

$m_1, m_2, m_3, d_0, d_1, d_2, d_3, d_4$

lstr

$m_0, m_1, m_2, m_3, d_0, d_1, d_2, d_3, d_4$

d_4

lstr

lstr

lstr

$\phi \phi \phi \phi m_0, m_1, m_2, m_3$

d_0

lstr

1

sec

rbc #1

beq exd

$\phi \phi \phi m_0, m_1, m_2, m_3, \phi$

ϕ

asl

lstr

2820

dnb...wr

asl asto

5

meus:

8/r 46

bcc

2, 3

lda # "D"

2

buc

2, 3

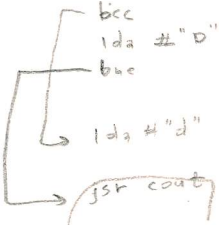
100: 442F

bu: 4446

len: 44444C

Cre: 14AC1812

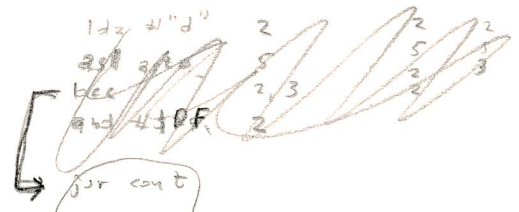
DNB 4444R



0101010101 = 24 and # DF

$D = 11000100 \leftarrow c=1$
 $d = 11100100 \leftarrow c=0$

5 5
 3 3
 2 3
 10 17
 10
 cycles



AC 44

94 AC

18 12
 18
 20

18:28

6:24 pm



1097 1092 1091 1090 1089 1088 1087 1086 1085 1084 1083 1082 1081 1080 1079 1078 1077 1076 1075 1074 1073 1072 1071 1070 1069 1068 1067 1066 1065 1064 1063 1062 1061 1060 1059 1058 1057 1056 1055 1054 1053 1052 1051 1050 1049 1048 1047 1046 1045 1044 1043 1042 1041 1040 1039 1038 1037 1036 1035 1034 1033 1032 1031 1030 1029 1028 1027 1026 1025 1024 1023 1022 1021 1020 1019 1018 1017 1016 1015 1014 1013 1012 1011 1010 1009 1008 1007 1006 1005 1004 1003 1002 1001 1000

April 20

5 5
 01010101

1198, 1227

10 11 12 13 14 15 16 17

YES!

setfpos

~~lda #0~~
~~lda #0~~
lda fvt, x
sta vtab
jsr pascal
lda fht, x
~~sta fht~~
~~lda #0~~
rts

i: X:field #
o: pas, ~~htab~~ ~~(hmap)~~
↑ ~~set fvt, x~~
~~hmap~~

ldx #5
jsr setfpos

o: pas
~~hmap~~
~~x = (hmap) +~~
hmap

tx
~~lda htab~~
~~ldx #0~~
~~jsr out~~
~~sty htab~~
ldy #314
lda (curaddr), y
ldy htab
jsr out
sty htab
ldx #313
lda (curaddr), y
ldy htab
jsr out
~~hmap~~

setfpos: i: X (field #)
o: pas
a (hmap)

ldi buf+addr+1
 adc #2
 stb buf+addr+1

4

2

4

10

*

inc buf+addr+1 6

inc buf+addr+1 6

12

readir ldi # < dirbuf 2
 stb rdx+1 4
 ldi # > dirbuf 2
 stb rdx+2 4

[MLI CALL]

rdsk1 ldi # 5+2 2
 rdx ldi \$FFFF, x 4
 stb block-2, x 5
 inc 2
 cmp # 4 2
 bne rdx 2, 3 if br

10 ins

45

readir ldi # < dirbuf 2
 stb rdx+1 4
 stb rdx+1+1 4
 ldi # > dirbuf 2
 stb rdx+2 4
 inc rdx+2 4

[MLI CALL]

rdsk1 ldi # 5+2 2
 rdx ldi \$FFFF, x 4
 stb block 4
 inc 2
 ldi \$FFFF, x 2
 stb block+1 4

12 ins

40

readir ldi # < dirbuf 2
 stb addr 4
 ldi # > dirbuf 2
 stb addr+1 4

[MLI CALL]

rdsk1 ldi # 5+2 2
 ldi (addr), x 5
 stb block 4
 inc 2
 ldi (addr), x 5
 stb block+1 4

10 ins

34

COMMAND SYSTEM

Ad 8 E C F
2 d D A F D
4 C d 1 d 8

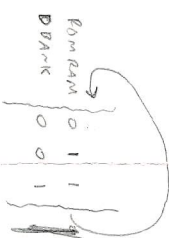
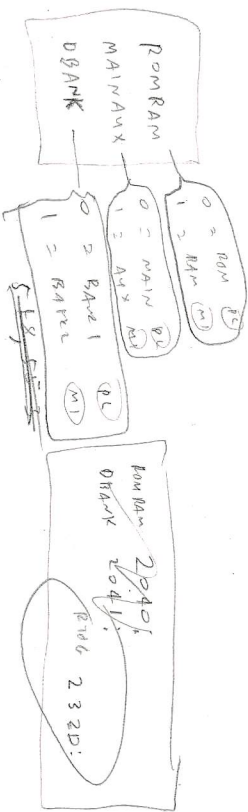
0 0 1
0 0 0

CURBAS
= F E F D

COMMAND SYSTEM

MAXIMUM OF 832 COMMANDS IN COMMAND/COMMANDS

CURBAS
VTAB
COPY
(MAIN) RAM
D BANK



A=80

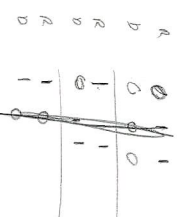
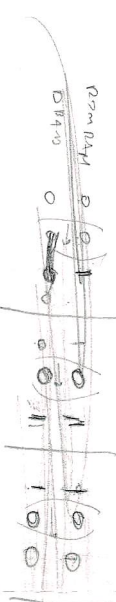
0 1 1 1 1
0 1 1 1 1
0 1 1 1 1

BIT: NVZ

233

855, 869

2 d



233:1C
233:24
233:24

CLC

LDA #1

ADC #87F

V=1



8

FE00
10
20
30

40
50
60
70
80
90

74

01110000

44

01110000

???

$MAV_{MAX} = 4\phi$

$MAV_{MAX} = 8\phi$

[illegible]

$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$
 $\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$
 $\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

100-EC

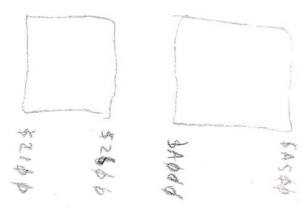
$$A = 1000 \text{ fph}$$

C | 0 0 0 0 0 0
 C | 0 0 0 0 0 0
 0 | 0 0 0 0 0 0
 0 | 0 0 0 0 0 0
 0 | 0 1 1 1 1 1

 0 | 0 1 1 1 1 1
 V C 0

COMMAND SYSTEM

MAC \$2100 LENGTH \$45
DEST \$A000



5A50

PART 1

FRONT END LOADER

MEN
COMMAND

~~FRONT END FAST LOADER~~

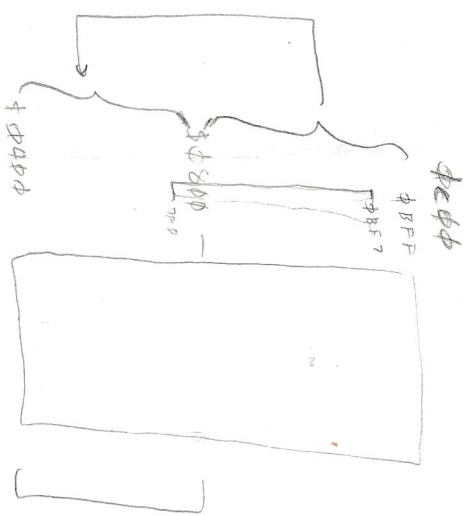
7FD: SMP \$4B97

828: \$3 \$1 \$8 1C 4 \$ 8 \$ \$ \$
\$4 \$428 (14 pairs)

\$4 \$4 \$1 \$4
(27 total)

865: LDA \$4428, Y
BEQ \$48A8
SRA \$FC
INY
LDA \$4428, Y
SRA \$EF
SY \$41
135

8A8:



2PG

EF: \$1

F3: 14 (offset of first 1/5 pair is 5000) use as variable

F1: ~~variable~~ Y510

F7: ~~variable~~ first track

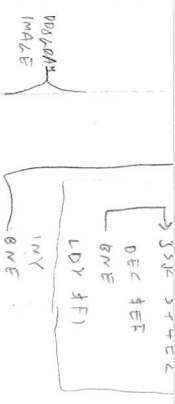
FB: \$4

FC: \$3

FD: \$1

FE: 12

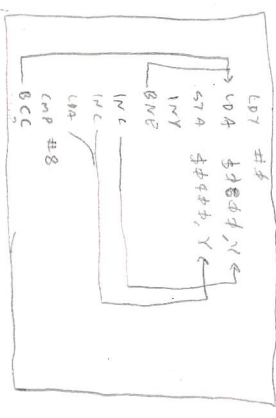
16\$



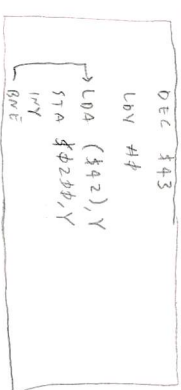
8RA



B77: LDA 50478
STA 3F7



B78: LDA 50478
STA 3F7



IMP 30465

B77
B78

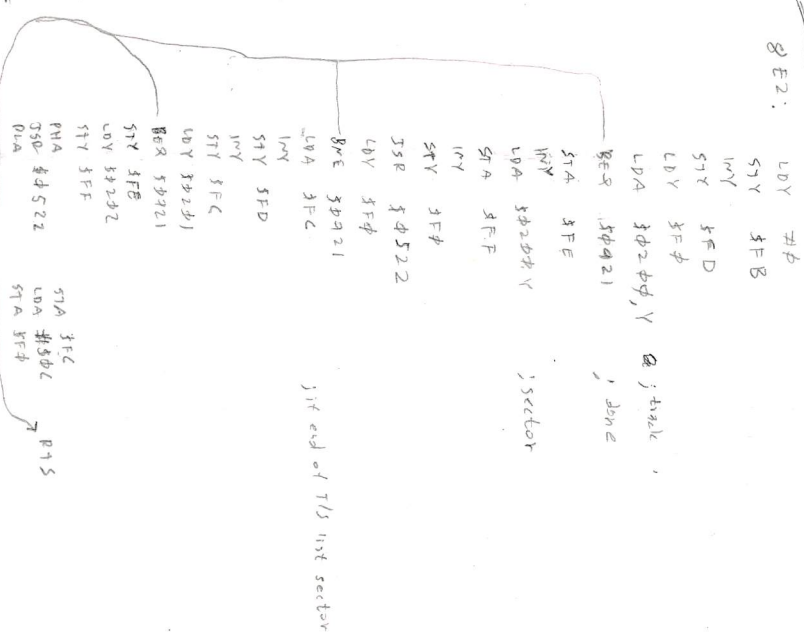
157

track number of first sector

30465 < 30465, 3BFF

move image of 7/5 1st first sector to 30200

8E2:



3A

722:

30522 read a sector?

495:

498:

4ED:

494:

494: